

npm Package Registry



- [Requirements](#)
 - [Configuring the package registry](#)
 - [Publish a package](#)
 - [Unpublish a package](#)
 - [Install a package](#)
 - [Tag a package](#)
 - [Search packages](#)
 - [Supported commands](#)
 - [Troubleshooting](#)
 - [Forgejo Actions Example](#)
-

Publish [npm](#) packages for your user or organization.

Requirements

To work with the npm package registry, you need [Node.js](#) coupled with a package manager such as [Yarn](#) or [npm](#) itself.

The registry supports [scoped](#) and unscoped packages.

The following examples use the `npm` tool with the scope `@test` and username `testuser`.

Configuring the package registry

To register the package registry with npm, you need to configure a new package source and provide the package owner's token.

```
npm config set {scope}:registry https://forgejo.example.com/api/packages/{owner}/npm/  
npm config set -- '//forgejo.example.com/api/packages/{owner}/npm/:_authToken' "{owner_token}"
```

NOTE: in the example below (`npm config set -- '//forgejo...`) the leading scheme, `https:`, is intentionally missing. It must not be included. The following is **incorrect**: `npm config set -- 'https://forgejo...` as the npm config only uses a URI fragment in the config. [Examples](#)

Parameter	Description
<code>scope</code>	The scope of the packages.
<code>owner</code>	The owner of the package.
<code>token</code>	The owner's personal access token .

For example:

```
npm config set @test:registry https://forgejo.example.com/api/packages/testuser/npm/  
npm config set -- '//forgejo.example.com/api/packages/testuser/npm/:_authToken'  
"{personal_access_token}"
```

or without scope:

```
npm config set registry https://forgejo.example.com/api/packages/testuser/npm/  
npm config set -- '//forgejo.example.com/api/packages/testuser/npm/:_authToken'  
"{personal_access_token}"
```

Publish a package

When publishing a package, npm uses configs and defaults if the arguments are not specified in the command or project. The following commands demonstrate a default publish, publishing to a specific registry, and a specific scope and registry:

```
npm publish  
npm publish --registry={SERVER_URL}/api/packages/{owner}/npm/  
npm publish --scope=@{scope} --registry={SERVER_URL}/api/packages/{owner}/npm/
```

You cannot publish a package if a package of the same name and version already exists. First, you must delete that version of the existing package, either in UI at `SERVER_URL/OWNER/-/packages/npm/PACKAGE_NAME/PACKAGE_VERSION/settings` or `unpublish` in CLI.

Unpublish a package

Delete a package by running the following command:

```
npm unpublish {package_name}[@{package_version}]
```

Parameter	Description
<code>package_name</code>	The package name.
<code>package_version</code>	The package version.

For example:

```
npm unpublish @test/test_package  
npm unpublish @test/test_package@1.0.0
```

Note: This behavior is different than the public npm repository. Once you delete or unpublish a package, you can re-publish a package with the same name and version immediately.

Install a package

To install a package from the package registry, execute the following command:

```
npm install {package_name}
```

Parameter	Description
<code>package_name</code>	The package name.

For example:

```
npm install @test/test_package
```

Tag a package

The registry supports [version tags](#), which can be managed by `npm dist-tag`:

```
npm dist-tag add {package_name}@{version} {tag}
```

Parameter	Description
<code>package_name</code>	The package name.
<code>version</code>	The version of the package.
<code>tag</code>	The tag name.

For example:

```
npm dist-tag add test_package@1.0.2 release
```

The tag name must not be a valid version. All tag names which are parsable as a version are rejected.

Search packages

The registry supports [searching](#) but does not support special search qualifiers like `author:forgejo`.

Supported commands

```
npm install
npm ci
npm publish
npm unpublish
npm dist-tag
npm view
npm search
```

Troubleshooting

Run these commands before the command that fails:

```
npm config set loglevel verbose    #The default log level is notice
npm config ls -l                   #Lists the full config, including defaults.  User config is
at the bottom.
```

The [npm CLI Docs](#) may help you find missing config settings required to be set prior to the failing command. Incorrect [Auth-related config](#) can silently/cryptically fail at the default log level.

Forgejo Actions Example

```
jobs:
  publish:
    name: Publish to Forgejo NPM registry
    runs-on: docker
    steps:
      - name: set npm config
        # See note above about using {server_uri_fragment}, i.e. '//code.forgejo.org'
        # DO NOT USE ${secrets.FORGEJO_TOKEN}, use the owner's token!
        run: |
          npm config set @scope:registry ${github.SERVER_URL}/api/packages/{owner}/npm/
          npm config set
            "${server_uri_fragment}/api/packages/{owner}/npm/:_authToken=${secrets.OWNER_TOKEN}"
      - name: Checkout repo
        uses: actions/checkout@v6
      - name: Install Dependencies
        run: npm install
      - name: Build
        run: npm run build
        # WARNING: The following commented step will override the default CA settings (null ==
        # only use known CAs) and prevent npm from communicating with the public npm registry
        # To undo it, add another step: 'run: npm config set cafile='
        #- name: Only if using a self-signed cert
        #   run: npm config set cafile {location of ca-cert.pem}
      - name: Publish to registry
        run: npm publish --scope=@scope --
          registry=${github.SERVER_URL}/api/packages/{owner}/npm/
```

When using Forgejo Actions with the NPM registry, you may have to use npm to obtain certain information. For instance, due to the structure of Forgejo's package storage, the published tarball is not stored in the `SERVER_URL/USER/-/packages/npm/PACKAGE_NAME/PACKAGE_VERSION/files/PACKAGE_ID/` location in the UI where it can be manually obtained. If you need to access the tarball directly, use this command to get a URL compatible with curl and other CLI tools that can run in Forgejo Actions.

```
npm view PACKAGE_NAME{@PACKAGE_VERSION} --registry={REGISTRY_URL} dist.tarball
```

📁 #3

📁 cocobook-sync 📁 2026-10-28 16:38:49 UTC

📁 cocobook-sync 📁 2026-10-29 01:09:31 UTC